



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/789,201	02/27/2004	Christopher W. Brumme	MS1-1771US	9134
22801	7590	10/31/2007	EXAMINER	
LEE & HAYES PLLC 421 W RIVERSIDE AVENUE SUITE 500 SPOKANE, WA 99201			NGUYEN, PHILLIP H	
		ART UNIT	PAPER NUMBER	
		2191		
		MAIL DATE	DELIVERY MODE	
		10/31/2007	PAPER	

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/789,201	BRUMME ET AL.
	Examiner	Art Unit
	Phillip H. Nguyen	2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 16 August 2007.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1,3-7,10,11,14-20,22,24-27,29-31 and 39-54 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1,3-7,10,11,14-20,22,24-27,29 and 39-51 is/are rejected.
- 7) Claim(s) 30,31 and 52-54 is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
 3) Information Disclosure Statement(s) (PTO/SB/08)
 Paper No(s)/Mail Date _____
- 4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date. _____
- 5) Notice of Informal Patent Application
 6) Other: _____

DETAILED ACTION

1. This action is in response to the amendment filed 8/16/2007.
2. Claims 1, 3-7, 10-11, 14-20, 22, 24-27, 29-31 and 39-54 remain pending and have been considered below.

Response to Amendment

3. Per Applicants' request, claims 1, 3-7, 10-11, 14-20, 22, 24-27, 29-31, 39-41, 48-49 and 51 are amended; claims 2, 8-9, 12-13, 21, 23 and 28 have been canceled.
4. The rejection to claims 1-6, 10-11, 22, 24-25, 29 and 39-40 under 35 USC § 112, second paragraph is withdrawn in view of applicants' amendment to these claims to clarify the discrepancies.
5. The rejection to claims 1-25 under 35 USC § 101 is withdrawn in view of applicants' amendment to claim 1 to clarify a computer storage media.

Response to Arguments

6. Applicant's arguments filed 8/16/2007 have been fully considered but they are not deemed persuasive.

Applicant asserts regarding claims 1, 7 and 14 that Lowney fails to teach the use of a rewrite list that identifies which rewriter is to be applied to rewrite a code unit and the sequence in which the rewrite manager applies the rewrites accordingly.

Examiner respectfully disagrees with the allegation as argued. First, It is unclear to examiner whether a rewriter list is actually a list that includes rewriters or it is just a software program that identifies and sequences the rewriter. Furthermore, a rewrite manager is also performing the same functions as the rewriter list ("whereby the rewrite manager identifies and sequences the appropriate rewriters by consulting the rewriter list"). Second, if a rewriter list identifies one rewriter then no need to sequence the rewriter since there is only one rewriter. Third, Lowney discloses, "The operating system controls the basic operations of the computer hardware and presents a framework for allowing other software applications to interface with the hardware" (see col. 1, lines 15-18). In other words, the operating system 20 in memory 18 manages the operation of the pre-optimizer 25 and the optimizer 30. In Lowney's approach, the operating system can be considered as a rewriter list and/or a rewrite manager because it manages the all the application inside memory 18.

Claim Objections

7. Claim 22 is objected to because of the following informalities: Since claim 21 has been canceled, claim 22 depends on claim 14. Appropriate correction is required.

Allowable Subject Matter

8. Claims 30-31 and 52-54 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Claim Rejections - 35 USC § 112

9. The following is a quotation of the second paragraph of 35 U.S.C. 112:
- The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.
10. Claims 1, 3-6, 26, 27, 29-31 and 39-40 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.
- Regarding claim 1 recites the limitation "a rewriter list...wherein the rewriter list identifies which rewriter...and provides the sequence the rewriters are applied..." There is insufficient antecedent basis for this limitation in the claim. Applicant further recites the limitation "...whereby the rewrite manager identifies and sequences the appropriate rewriters by consulting the rewriter list." It is unclear to examiner whether the rewriter list is the same as the rewrite manager. Both of them identify and sequence the rewriters. For examining purposes, examiner assumes they are the same. Claims 3-6 directly or indirectly depend on claim 1 and therefore suffer the same deficiency.
 - Regarding claim 26 recites the limitation "wherein the rewrite manager determines the identity and sequence..." There is insufficient antecedent basis for this limitation in the claim. For examining purpose, examiner assumes rewrite manager accessing a rewrite list that identifies which rewriter to apply to rewrite the code unit and provides the sequence the rewriters are applied to the code unit.

- Regarding claim 41 recites the limitations "...a rewrite list that identifies which rewriter to apply...and provides that sequence the rewriters..." It is unclear whether the rewriter list is the same as rewrite manager. Both of them identify and sequence the rewriter.

Claim Rejections - 35 USC § 102

11. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

12. Claims 1, 3-7, 10, 11, 14, 17, 18, 20, 22, 24 and 25 are rejected under 35 U.S.C. 102(b) as being anticipated by Lowney et al. (United States Patent No.: US 6,324,689 B1).

As per claims 1, 7, 14:

Lowney discloses:

- a code unit having executable instructions (see at least col. 3, line 64-65 "executable files (shown labeled EXE)... are software application"; also see FIG. 3);

- one or more rewriters, wherein each rewriter modifies the code unit in a unique way (see at least col. 4, line 5-6 “**... optimizer routine 30 operate ... to re-write the executables 21 and 22**”);
- a rewriter list (see at least FIG. 1, item 20 “**operating system**”) identifying at least one rewriter, wherein the rewriter list identifies which rewriter to apply to rewrite the code unit and provides the sequence the rewriters are applied to the code unit (see at least col. 1, lines 15-18 “**the operating system controls the basic operations of the computer hardware and presents a framework for allowing other software applications to interface with the hardware**” – *the operating system identifies the optimizer routine 30 by allowing it to rewrite the executable files*); and
- a rewrite manager (see at least FIG. 1, item 20 “**operating system**”) having executable instructions configured to access one or more rewriter and execute the one or more rewriters against the code unit to generate a rewritten code unit, whereby the rewrite manager identifies and sequences the appropriate rewriters by consulting the rewriter list (see at least col. 1, lines 15-18 “**the operating system controls the basic operations of the computer hardware and presents a framework for allowing other software applications to interface with the hardware**” – *in other words, the operating system control the operating of the optimizer routine 30*).

Art Unit: 2191

As per claim 3:

Lowney further discloses:

- a cache, the rewrite manager further configured to store the rewritten code unit in the cache (see at least col. 3, line 60-61 "**during operation, as instructions and data of the software are executed...copied into the cache 16**").

As per claim 4:

Lowney further discloses:

- wherein the rewriter list is selected from a group comprising:
 - o a list of one or more custom attributes within the code unit, a list within a security policy, a list within an installation tool, a list within a configuration file, and a list within an XML (Extensible Markup Language) file (see at least **FIG. 1**).

As per claim 5:

Lowney further discloses:

- wherein the rewrite manager is a module selected from a group comprising:
 - o a stand-alone module; an operating system module; an execution environment module; a JIT compiler module; a source code compiler module; and an installation tool configured to install the code unit (see at least **FIG. 1**).

As per claims 6 and 25:

Lowney discloses:

- wherein the computer comprising:
 - o a developer computer configured to create the code unit; an intermediate computer configured to deploy the code unit; and a deployment computer configured to execute the code unit (see at least col. 3, line 49 "**a computer system**").

As per claim 10:

Lowney discloses:

- wherein the initiating comprises: identifying an environment in which the transformation should be implemented (see at least col. 3, line 49-55 "**a computer system 10...includes a central processing unit (CPU)...memory 18...a cache 16... a processor 14..."**); and
- initiating the transformation only if the code unit experiences the identified environment (**the re-write process is performed for the computer system 10 is identified**).

As per claim 11:

Lowney further discloses:

- wherein identifying an environment comprises:

- identifying a source code compilation environment (see at least col. 3, line 49-55 “**a computer system 10...includes a central processing unit (CPU)...memory 18...a cache 16... a processor 14...**”);
- identifying a pre-execution-compilation environment (see at least col. 3, line 49-55 “**a computer system 10...includes a central processing unit (CPU)...memory 18...a cache 16... a processor 14...**”);
- identifying a compilation-on-installation environment (see at least col. 3, line 49-55 “**a computer system 10...includes a central processing unit (CPU)...memory 18...a cache 16... a processor 14...**”);
- identifying an execution environment (see at least col. 3, line 49-55 “**a computer system 10...includes a central processing unit (CPU)...memory 18...a cache 16... a processor 14...**”); and
- identifying an installation environment (see at least col. 3, line 49-55 “**a computer system 10...includes a central processing unit (CPU)...memory 18...a cache 16... a processor 14...**”).

As per claim 17:

Lowney discloses:

- computer-executable instructions configured for storing the rewritten code unit in a cache (see at least col. 3, line 60-61 “**during operation, as instructions and data of the software are executed...copied into the cache 16**”).

As per claim 18:

Lowney discloses:

- receiving an instruction to execute the code unit (see at least col. 4, line 5-6 "**pre-optimizer routine 25 optimizer routine 30 operate together to re-write the executables 21 and 22**");
- recognizing that the code unit has been rewritten (see at least col. 3, line 60-61 "**during operation, as instructions and data of the software are executed...copied into the cache 16**" – by **recognizing the executed code has been rewritten, copying the executed code into the cache 16**);
- loading the rewritten code unit from the cache (see at least col. 3, line 60-61 "**during operation, as instructions and data of the software are executed...copied into the cache 16**"); and
- executing the rewritten code unit (see at least FIG. 1, **processor 14 is executing the code in cache 16**).

As per claim 20:

Lowney discloses:

- wherein the at least one rewriter is a plurality of rewriters (see at least col. 4, line 5 "**optimizer routine 30**" – *at least one rewriter*), the one or more computer-readable media comprising further computer-executable instructions configured for sequencing the plurality of rewriters to rewrite the code unit in a particular

rewrite order (**if only optimizer routine then no need to sequence the optimizer**).

As per claim 22:

Lowney further discloses:

- wherein the accessing comprises accessing the rewriter list in a location selected from a group comprising:
 - o the code unit, a separate file associated with the code; and a system policy (see at least **FIG. 1**).

As per claim 24:

Lowney further discloses:

- embodies in a tool selected from a group comprising:
 - o a source code compiler, an installation tool, a managed execution environment, a stand-alone rewrite management tool and a JIT (Just in Time) compiler (see at least col. 3, line 67 "**a compiler**"; also see at least **FIG. 1**).

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

14. Claims 26, 27, 29, 39-41, 44-45 and 47-51 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lowney et al. (United States Patent No.: US 6,324,689 B1).

As per claim 26:

Lowney discloses:

- a code unit (see at least col. 3, line 64-65 "**executable files (shown labeled EXE)... are software application**"; also see FIG. 3);
- a composable set of rewriters, each rewriter configured to rewrite the code unit in a unique manner;
- a rewrite manager configured to identify one or more rewriters from the composable set of rewriters, sequence the rewriters and execute the identified one or more rewriters against the code unit, wherein the rewrite manager determines the identity and sequence by accessing a rewrite list that identifies which rewriter to apply to rewrite the code unit and provides the sequence the rewriters are applied to the code unit (see at least col. 1, lines 15-18 "**the operating system controls the basic operations of the computer hardware and presents a framework for allowing other software**

- applications to interface with the hardware**" – *the operating system identifies the optimizer routine 30 by allowing it to rewrite the executable files*); and
- a rewritten code unit generated by executing the identified one or more rewriters against the code unit (see at least col. 5, line 44-45 "**optimization tool re-writes the executable ...**").

Lowney does not explicitly disclose:

- a composable set of rewriters, each rewriter configured to rewrite the code unit in a unique manner.

However, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Lowney's approach to include multiple optimizers. One would have been motivated to modify Lowney's approach because it would give the system or the users an option to select which optimizer (rewriter) to rewrite the executable files.

As per claim 27:

Lowney further discloses:

- a rewrite cache, the rewrite manager further configured to store the rewritten code unit in the rewrite cache (see at least col. 3, line 60-61 "**during operation, as instructions and data of the software are executed...copied into the cache 16**").

Art Unit: 2191

As per claim 29:

Lowney further discloses:

- wherein the rewriter list is a component selected from a group comprising:
 - o a list of rewriters in the code unit; a list of rewriters in a stand-alone file; a list of rewriters in a security policy; and a list of rewriters in an installation tool (**optimization tool is stand-alone tool**).

As per claim 39:

Lowney further discloses:

- wherein the rewrite manager is a component selected from a group comprising:
 - o a stand-alone rewrite module; a rewrite module configured as part of an operating system; a rewrite module configured as part of an installation tool; and a rewrite module configured as part of a security policy (**pre-optimization is stand-alone tool**).

As per claim 40:

Lowney further discloses:

- a development computer configured to develop the code unit; an intermediate computer configured to install the code unit; and a deployment computer configured to execute the code unit (see at least col. 3, line 49 "**a computer system**").

As per claim 41:

Lowney discloses:

- receiving an executable code unit (see at least col. 3, line 58-59 “**software that executes on the computer system 10 is shown stored in the memory 18**”);
- determining that the code unit needs to be rewritten (see at least col. 5, line 31-33 “**analyzes the executable to identify longwords as either instructions or data**”);
- determining one or more rewriters to rewrite the code unit (see at least col. 4, line 5 “**optimizer routine 30**”), wherein the rewriter is determined by a rewrite manager that accesses a rewrite list that identifies which rewriter to apply to rewrite the code unit and provides the sequence the rewriters are applied to the code unit (see at least col. 1, lines 15-18 “**the operating system controls the basic operations of the computer hardware and presents a framework for allowing other software applications to interface with the hardware**” – *the operating system identifies the optimizer routine 30 by allowing it to rewrite the executable files*); and
- running the one or more rewriters against the code unit to generate a rewritten code unit (see at least col. 5, line 44-45 “**optimization tool re-writes the executable ...**”).

Lowney does not explicitly disclose:

- a rewrite list.

Art Unit: 2191

However, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Lowney's approach to include a list of rewriters. One would have been motivated to modify because it would give the system or user an option to select an optimizer to rewrite the executable files.

As per claim 44:

Lowney further discloses:

- storing the rewritten code unit in a cache (see at least col. 3, line 60-61 "**during operation, as instructions and data of the software are executed...copied into the cache 16**".)

As per claim 45:

Lowney further discloses:

- receiving an instruction to execute the code unit (see at least col. 4, line 5-6 "**pre-optimizer routine 25 optimizer routine 30 operate together to re-write the executables 21 and 22**");
- recognizing that the rewritten code unit is stored in the cache (see at least col. 3, line 60-61 "**during operation, as instructions and data of the software are executed...copied into the cache 16**" – by recognizing the executed code has been rewritten, copying the executed code into the cache 16);

- loading the rewritten code unit from the cache (see at least col. 3, line 60-61
"during operation, as instructions and data of the software are executed...copied into the cache 16"); and
- executing the rewritten code unit (see at least FIG. 1, **processor 14 is executing the code in cache 16**).

As per claim 47:

Lowney further discloses:

- sequencing the one or more rewriters to rewrite the code unit in a particular rewrite order (**since only one optimizer in Lowney's approach, sequence is no needed**).

As per claim 48:

Lowney further discloses

- accessing the rewriter list; and setting the rewrite order according to the rewriter list (**It is inherent in Lowney's approach if including a list that includes a plurality of rewriters**).

As per claim 49:

Lowney further disclose:

Art Unit: 2191

- wherein the determining one or more rewriters comprises reading the rewriter list

(It is inherent if Lowney's approach includes a list that has a plurality of rewriters).

As per claim 50:

Lowney further discloses:

- wherein the reading comprises reading the rewriter list in a location selected from the group comprising:
 - o the executable code and a separated file associated with the executable code unit (it is inherent if Lowney's approach includes a list that includes a plurality of rewriter).

As per claim 51:

Lowney discloses:

- receiving a code unit (see at least col. 3, line 58-59 "**software that executes on the computer system 10 is shown stored in the memory 18**");
- determining that the code unit is to be rewritten by a rewriter rewritten (see at least col. 5, line 31-33 "**analyzes the executable to identify longwords as either instructions or data**");
- determining if the code unit and the rewriter are trusted (**instructions and data of the software and optimization tool are always trusted software**);

- running the rewriter against the code unit to generate a rewritten code unit if the code unit and the rewriter are trusted (see at least col. 5, line 44-45 “**optimization tool re-writes the executable ... ”**); and
- storing the rewritten code unit in a cache (the executable code was originally copied into the cache a (see at least col. 3, line 60-61 “**during operation, as instructions and data of the software are executed...copied into the cache 16”**).

Lowney does not explicitly disclose:

- wherein the rewriter is determined by a rewrite manager that accesses a rewrite list that identifies which rewriter to apply to rewrite the code unit and provides the sequence the rewriters are applied to the code unite.

However, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Lowney’s approach to include a list of rewriters. One would have been motivated to modify because it would give the system or user an option to select an optimizer to rewrite the executable files.

15. Claims 15, 16, 19, 42, 43 and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lowney et al. (United States Patent No.: US 6,324,689 B1), in view of Cotugno et al. (United States Patent No.: US 6,353,887 B1).

As per claims 15 and 42:

Lowney does not explicitly disclose:

- computer-executable instructions configured for verifying trustworthiness of the code unit and the at least one rewriter prior to the executing.

However, Cotugno discloses an analogous way of distributing applications:

- computer-executable instructions configured for verifying trustworthiness of the code unit and the at least one rewriter prior to the executing (col. 20, line 53-56)
"One should not trust files that are supposedly wrapped with digital signatures under conditions when the system does not either attempt to verify the signature or fails to do so".

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Lowney's approach to include a digital signature for verifying the files. One would have been motivated to include a digital signature for verifying trustworthy of the files and verifying that the files have not been tampered or altered during it transmission (see Cotugno at least col. 20-21).

As per claims 16 and 43:

Cotugno discloses:

- wherein the verifying comprises authenticating a digital signature (see at least col. 18, line 20-21 "**authenticate the signature and verifying that the file has not been tampered...**").

As per claims 19 and 46:

Lowney does not explicitly disclose:

Art Unit: 2191

- generating a digital signature for the rewritten code unit; and
- associating the digital signature with the rewritten code unit.

However, Cotugno discloses an analogous way of distributing files.

- generating a digital signature for the rewritten code unit (see at least col. 18, line 23 "**Digital signature "creation" is based on the data within a file and a private key**"); and
- associating the digital signature with the rewritten code unit (see at least col. 18, line 23-26 "**Digital signature "creation" is based on the data within a file and a private key digital signature "authentication" is based on the same data of the file and a corresponding public key**").

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to modify Lowney's approach to include a digital signature for verifying the files. One would have been motivated to create a digital signature for verifying trustworthy of the files and verifying that the files have not been tampered or altered during its transmission (see Cotugno at least col. 20-21).

Conclusion

16. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Thursday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Application/Control Number: 10/789,201
Art Unit: 2191

Page 23

PN
10/28/2007



WEI ZHEN
SUPERVISORY PATENT EXAMINER